Rotman

INTRO TO R PROGRAMMING

R Tutorial (RSM456) – Session 2

January 14, 2025 Prepared by Jay Cao / TDMDAL

Website: https://tdmdal.github.io/r-intro-2025-winter/

Rotman School of Management
UNIVERSITY OF TORONTO

Plan

- Descriptive statistics
- T-test
- Linear regression

Description Statistics

- `mean(df$col1), median(df$col1), var(df$col1), sd(df$col1)`
- `cor(df$col1, df$col2) or cor(df[c("col1", "col2", "col3")])`
- Simple histogram: `hist(df$col1)`
- Simple scatter plot: `plot(df$col1, df$col2)`
- Summary statistics for a dataframe's numeric columns: `summary(df)`

T-test

- A hypothesis test for evaluating means of one or two populations

	One-sample t-test	Two-sample t-test	Paired t-test
Purpose	Decide if the population mean is equal to a specific value or not	Decide if the population means for two different groups are equal or not	Decide if the difference between paired measurements for a population is zero or not
Example	Mean heart rate of a group of people is equal to 65 or not	Mean heart rates for two groups of people are the same or not	Mean difference in heart rate for a group of people before and after exercise is zero or not

- Note: t-test in regression analysis

Source and full table: https://www.jmp.com/en_ca/statistics-knowledge-portal/t-test.html

T-test in R, `t.test()`

- One-sample

```
t.test(x = heart_rate, mu = 71, alternative = "two.sided")
```

- Two-sample

```
t.test(x = heart_rate_group1, y = heart_rate_group2,
                alternative = "two.sided")
```

- Paired

```
t.test(x = heart_rate_before_ex, y = heart_rate_after_ex,
            alternative = "two.sided", paired = TRUE)
```

R t-test() document: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/t.test.html

Linear Regression - Housing Price & Clean Air

- Manipulate data
  - Load data
  - Create new columns
  - Filter columns and rows
- Build models
  - Multiple linear regressions
- Report and graph
  - Plot a few graphs
  - Report regression results

```
Obs:   506

1. price      median housing price, $
2. crime      crimes committed per capita
3. nox        nitrous oxide, parts per 100 mill.
4. rooms      avg number of rooms per house
5. dist       weighted dist. to 5 employ centers
6. radial     accessibiliy index to radial hghwys
7. proptax    property tax per $1000
8. stratio    average student-teacher ratio
9. lowstat    % of people 'lower status'
```

Choice 1: Use Only Base R packages

- Manipulate data
  - Load data (`read.csv()`)
  - Create new columns (`base R data frame manipulation`)
  - Filter columns and rows (`base R data frame manipulation`)

- Build models
  - Multiple regression (`lm()` from `stats` library in R base)

- Report and graph
  - Base R plot system, `plot()`
  - Base R `summary()` function

- A Note on Predictive Analysis
  - Train and test (or validation) split
  - Predict on test data and obtain evaluation measures of interest

Choice 2: Use some non-Base-R Packages

- Manipulate data (`tidyverse` eco-system)
  - Load data (`read_csv()` from the `readr`)
  - Create new columns (`mutate()` from `dplyr`)
  - Filter columns and rows (`select()` and `filter()` from `dplyr`)

- Build models
  - Multiple regression (`lm()` from `stats` library in R base)

- Report and graph
  - Graph using ggplot2 and some of its extensions
  - Build a publication-ready table (`huxreg()` from `huxtable` library)

Load a CSV file

- ***Choice1***: `read.csv()` from Base R's utils library (load into dataframe)

```
read.csv(file)
```

```
e.g. hprice <- read.csv("hprice.csv")
```

- ***Choice 2***: `read_csv()` from `tidyverse`'s `readr` library (load into tibble/dataframe)

```
read_csv(file)
```

```
e.g. hprice <- read_csv("hprice.csv")
```

Data Frame Manipulation – Base R vs dplyr

Data Operation	*Choice 1*: Base R	*Choice 2*: dplyr

Filter rows based on conditions	`df[which(df$x > 0), , drop = FALSE], or`
	
	`subset()`
		`filter(df, x > 0)`

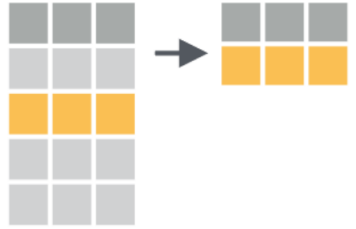
Create a new column variable (from other column variables)	`df$z <- df$x + df$y, or`
	
	`df["z"] <- df["x"] + df["y"], or`
	
	`transform()`
		`mutate(df, z = x + y)`

Select column variables	`df[c("x", "y")], or`
	
	`subset()`
		`select(df, x, y)`

…	…	…

Source: https://dplyr.tidyverse.org/articles/base.html

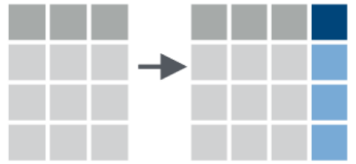
Choice 1 - Data Manipulation: Base R



- Filter observations (rows):

my_df[which(cond), , drop = FALSE]

```
e.g., hprice_reg <- hprice[which(hprice$price > 20000), , drop = FALSE]
```



- Create new variables (columns):

my_df["new_col"] <- expression (involving other cols)

```
e.g., hprice_reg["lprice"] <- log(hprice_reg["price"])
```

- Select variables (columns):

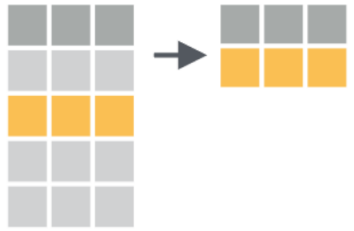
my_df[c("col1", "col2")]

```
e.g., hprice_reg <- hprice_reg[c("lprice", "rooms")]
```

Ref. dplyr data wrangling cheat sheet

Choice 2 - Data Manipulation: `dplyr` basics

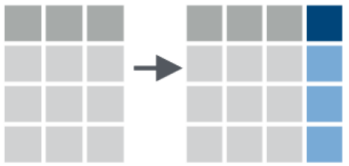
- Filter observations (rows): **`filter()`**



`filter(my_df, condition1, …)`

```
e.g., hprice_reg <- filter(hprice, price > 20000)
```

- Create new variables (columns): **`mutate()`**



`mutate(my_df, new_var1 = expression1, …)`

```
e.g., hprice_reg <- mutate(hprice_reg, lprice = log(price))
```

- Select variables (columns): **`select()`**

`select(my_df, var1, …)`

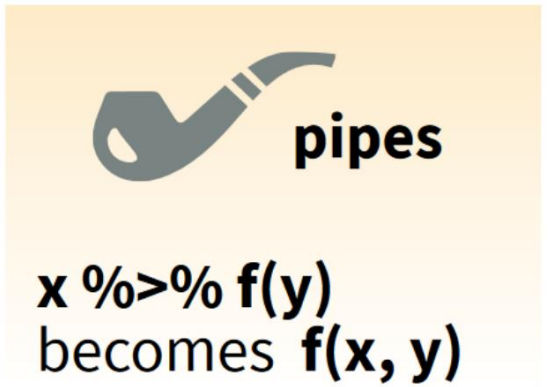
```
e.g., hprice_reg <- select(hprice_reg, lprice, rooms)
```

Ref. dplyr data wrangling cheat sheet

Choice 2 - Data Pipe (%>%) with `dplyr`

```
hprice_reg <- filter(hprice, price > 20000)
hprice_reg <- mutate(hprice_reg, lprice = log(price))
hprice_reg <- select(hprice_reg, lprice, rooms)
```

```
hprice_reg <- hprice %>%
  filter(price > 20000) %>%
  mutate(lprice = log(price)) %>%
  select(lprice, rooms)
```



Ref. dplyr data wrangling cheat sheet

Regression

- Multiple regressions: `lm()` from `stats` library in base R

```
my_model <- lm(y ~ x1 + x2, data)
```

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon_i$$

```
my_model <- lm(y ~ x1 + x2 + I(x1 * x2), data)
```

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon_i$$

- Regression result summary: `summary()`

Ref. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/formula.html
Note: See appendix for more on R regression formula

Report

- ***Choice 1***: Summary table of `lm()` (Base R)

```
summary(my_model)
```

- ***Choice 2***: publication-ready table: `huxreg()` from `huxtable` library

```
huxtable(my_model1, my_model2, …)
```

Ref. https://hughjonesd.github.io/huxtable/huxreg.html

Read the Regression Report

```
Call:

lm(formula = lprice ~ lnox + rooms + I(rooms^2) +
stratio, data = hprice_reg)


Residuals:

     Min       1Q   Median       3Q      Max

-0.67205 -0.11678  0.01795  0.11597  0.59801


Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept) 13.17845    0.47124  27.965  < 2e-16 ***

lnox        -0.58449    0.04594 -12.724  < 2e-16 ***

rooms       -0.69766    0.14221  -4.906 1.30e-06 ***

…
```

```
…

I(rooms^2)   0.07211    0.01129   6.385 4.29e-10 ***

stratio     -0.03929    0.00426  -9.223  < 2e-16 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.1833 on 448 degrees of freedom

Multiple R-squared:  0.6188, Adjusted R-squared:  0.6154

F-statistic: 181.8 on 4 and 448 DF,  p-value: < 2.2e-16
```

Interpret Regression Result (Coefficients)

- $y = \hat{\beta}_0 + \widehat{\boldsymbol{\beta}}_{\mathbf{1}} x_1 + \hat{\beta}_2 x_2$ (x_1 is continuous)
- $y = \hat{\beta}_0 + \widehat{\boldsymbol{\beta}}_{\mathbf{1}} x_1 + \hat{\beta}_2 x_2$ (x_1 is categorical, say, 0 or 1)

- $\log(y) = \hat{\beta}_0 + \widehat{\boldsymbol{\beta}}_{\mathbf{1}} x_1 + \hat{\beta}_2 x_2$ (y is log-transformed)
- $y = \hat{\beta}_0 + \widehat{\boldsymbol{\beta}}_{\mathbf{1}} \log(x_1) + \hat{\beta}_2 x_2$ (x_1 is log-transformed)
- $\log(y) = \hat{\beta}_0 + \widehat{\boldsymbol{\beta}}_{\mathbf{1}} \log(x_1) + \hat{\beta}_2 x_2$ (y and x_1 are log-transformed)

- $y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1 x_2$ (an interactive term)

Ref. https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faqhow-do-i-interpret-a-regression-model-when-some-variables-are-log-transformed/

A Note on Predictive Analysis

- Causal vs predictive analysis

- Training and test (validation) data split

- Three Steps
  1. randomly split the data into training and test set.
  2. train/estimate a model on training set.
  3. Evaluate the estimated model on test set, i.e., predict on the test set, and obtain evaluation measures of interest.

Appendix: R Regression Formula - 1

my_df

y	x1	x2	x3
18	8	307	130
16	8	304	150
…	…	…	…

lm() Regression Formula	Regression Formula

`lm(formula = y ~ x1 + x2 + x3, data = my_df)`
 	
`lm(formula = y ~ ., data = my_df)`
	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$

`lm(formula = y ~ . - x3, data = my_df)`
 	
`lm(formula = y ~ x1 + x2, data = my_df)`
	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$

`lm(formula = y ~ 0 + x1 + x2 + x3, data = my_df)`	$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$
…	…

Source: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/formula.html

Appendix: R Regression Formula - 2

my_df

y	x1	x2	x3
18	8	307	130
16	8	304	150
…	…	…	…

lm() Regression Formula	Regression Formula

`lm(formula = y ~ x1 * x2, data = my_df)`
 	
`lm(formula = y ~ x1 + x2 + x1:x2, data = my_df)`
 	
`lm(formula = y ~ x1 + x2 + I(x1 * x2), data = my_df)`
	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$

`lm(formula = y ~ x1 + I(x1^2), data = my_df)`	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \epsilon$
`lm(formula = y ~ x1 + log(x2), data = my_df)`	$Y = \beta_0 + \beta_1 X_1 + \beta_2 \ln(X_2) + \epsilon$
…	…

Source: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/formula.html